pysimavr Documentation

Release 0.0.1

ponty

July 26, 2011

CONTENTS

1 Basic usage 2

2 Installation 3
- 2.1 General . . . 3
- 2.2 Ubuntu . . . 3
- 2.3 Uninstall . . . 4

3 Usage 5

4 File hierarchy 6

5 API 7
- 5.1 low level interface . . . 7
- 5.2 high level interface . . . 13

6 Development 15
- 6.1 Tools . . . 15
- 6.2 Install on ubuntu . . . 15
- 6.3 Tasks . . . 16

7 Indices and tables 17

Python Module Index 18

Index 19

pysimavr

Date July 26, 2011

PDF pysimavr.pdf

Contents:

pysimavr is a python wrapper for simavr which is AVR and arduino simulator

Links:

- home: https://github.com/ponty/pysimavr
- documentation: http://ponty.github.com/pysimavr

Features:

- python wrapper using swig
- object oriented interface on top of the generated interface
- maximum speed can be real-time
- serial communication
- check simavr documentation

Known problems:

- Python 3 is not supported
- tested only on linux
- low sound quality
- more test needed
- PWM simulation is not real-time
- missing PWM modes
- a lot of messages on stdout

CHAPTER

ONE

BASIC USAGE

```
>>> from pysimavr.avr import Avr
>>> avr=Avr(mcu='atmega48',f_cpu=8000000)
>>> firmware = Firmware('lcd.elf')
>>> avr.load_firmware(firmware)

>>> from pysimavr.serial import ArduinoSimSerial
>>> print ArduinoSimSerial(snippet='Serial.print("hello!");').serial()
hello!
```

CHAPTER

TWO

INSTALLATION

check simavr doc: http://gitorious.org/simavr/pages/GetStarted

ignore these in simavr doc:

- OpenGl (freeglut)
- gcc-avr
- avr-libc
- make

2.1 General

- install python
- install setuptools
- install swig (for source build only)
- install header files and a static library for Python (for source build only)
- install a compiler (for source build only)
- install elf library
- install the program:

  ```
  # as root
  easy_install pysimavr
  ```

2.2 Ubuntu

```
sudo apt-get install python-setuptools
sudo apt-get install swig
sudo apt-get install python-dev
sudo apt-get install gcc
sudo apt-get install libelf-dev
sudo easy_install pysimavr
```

2.3 Uninstall

first install pip:

```
# as root
pip uninstall pysimavr
```

CHAPTER

THREE

USAGE

pysimavr.examples.simple:

```
from pysimavr.avr import Avr

avr=Avr(mcu=’atmega48’,f_cpu=8000000)
avr.step(1)
print avr.pc
```

```
$ python -m pysimavr.examples.simple
Starting atmega48 - flashend 0fff ramend 02ff e2end 00ff
atmega48 init
2
```

pysimavr.examples.hello:

```
from pysimavr.serial import ArduinoSimSerial

print ArduinoSimSerial(snippet=’Serial.print("hello!");’).serial()
```

```
$ python -m pysimavr.examples.hello
Note: avr_register_io_write(0056): Installing muxer on register.
Note: avr_register_io_write(0056): Installing muxer on register.
Loaded 2956 .text
Loaded 26 .data
Starting atmega128 - flashend 1ffff ramend 10ff e2end 0fff
atmega128 init
uart_udp_init bridge on port 4321
avr_timer_configure-0 TOP 976.56Hz = 16384 cycles
avr_timer_configure-0 B 3472.22Hz = 4608 cycles
avr_timer_configure-0 C 3472.22Hz = 4608 cycles
avr_timer_configure-1 TOP 30.52Hz = 524288 cycles
avr_timer_configure-1 TOP 3.81Hz = 4194304 cycles
avr_timer_reconfigure-1 unsupported timer mode wgm=1 (0)
avr_timer_configure-2 TOP 244.14Hz = 65536 cycles
avr_timer_configure-2 B 868.06Hz = 18432 cycles
avr_timer_configure-2 C 868.06Hz = 18432 cycles
avr_timer_reconfigure-2 unsupported timer mode wgm=1 (0)
avr_timer_configure-3 TOP 30.52Hz = 524288 cycles
avr_timer_configure-3 TOP 3.81Hz = 4194304 cycles
avr_timer_reconfigure-3 unsupported timer mode wgm=1 (0)
ADC Start AREF 0 AVCC 5000
UART-0 configured to 00cf = 4807 bps, 5 data 1 stop
Roughtly 1666 usec per bytes
hello!
```

CHAPTER

FOUR

FILE HIERARCHY

```
|-docs                   sphinx documentation
|---_build               generated documentation
|-pysimavr               main python package, high level classes
|---examples             examples
|---swig                 all swig files (simavr and parts)
|-----cores              copy from simavr
|-----include            copy from simavr
|-------avr              copy from avr-libc
|-----parts              some electronic parts in c
|-----sim                copy from simavr
|-tests                  unit tests
```

CHAPTER

FIVE

API

There are 2 interfaces:

- pysimavr.swig.*: low level, generated by swig
- pysimavr.*: high level classes, they can redirect function calls to low level interface. Example: Avr class (high level) has all properties and methods of avr_t class (low level) automatically.

5.1 low level interface

class `pysimavr.swig.ac_input.`**`ac_input_t`**

`avr`

`irq`

`value`

class `pysimavr.swig.hd44780.`**`hd44780_t`**

`avr`

`cursor`

`datapins`

`flags`

`h`

`irq`

`pinstate`

`readpins`

`vram`

`w`

class `pysimavr.swig.inverter.`**`inverter_t`**

`avr`

`irq`

out

class pysimavr.swig.ledrow.**ledrow_t**

avr

irq

pinstate

pinstate_changed

class pysimavr.swig.sgm7.**sgm7_t**

avr

digit_count

digit_pin

digit_port

digit_segments

digit_segments_changed

irq

pinstate

segment_pin

segment_port

class pysimavr.swig.simavr.**avr_io_t**

avr

dealloc

ioctl

irq

irq_count

irq_ioctl_get

irq_names

kind

next

reset

class pysimavr.swig.simavr.**avr_ioport_getirq_t**

bit

irq

class pysimavr.swig.simavr.**avr_ioport_state_t**

ddr

name

pin

port

class pysimavr.swig.simavr.**avr_ioport_t**

io

name

pcint

r_ddr

r_pcint

r_pin

r_port

class pysimavr.swig.simavr.**avr_irq_pool_t**

count

irq

class pysimavr.swig.simavr.**avr_irq_t**

flags

hook

irq

name

pool

value

class pysimavr.swig.simavr.**avr_kind_t**

make

names

class pysimavr.swig.simavr.**avr_symbol_t**

addr

symbol

class pysimavr.swig.simavr.**avr_t**

aref

avcc

codeend

cycle

cycle_timer

cycle_timer_map

data

e2end

eind

flash

flashend

frequency

fuse

gdb

gdb_port

i_shadow

init

io

io_port

io_shared_io

io_shared_io_count

irq_pool

mmcu

next_cycle_timer

pc

pending

pending_wait

ramend

rampz

reset

run

signature

sleep

special_deinit

special_init

sreg

state

trace

trace_data

vcc

vcd

vector

vector_size

class pysimavr.swig.simavr.**avr_t_cycle_timer**

param

timer

when

class pysimavr.swig.simavr.**avr_t_io**

irq

r

w

class pysimavr.swig.simavr.**avr_t_io_r**

c

param

class pysimavr.swig.simavr.**avr_t_io_shared_io**

io

used

class pysimavr.swig.simavr.**avr_t_io_shared_io_io**

c

param

class pysimavr.swig.simavr.**avr_t_io_w**

c

param

class pysimavr.swig.simavr.**avr_trace_data_t**

codeline

old

old_pci

touched

class pysimavr.swig.simavr.**avr_trace_data_t_old**

pc

sp

class pysimavr.swig.simavr.**avr_vcd_log_t**

signal

value

when

class pysimavr.swig.simavr.**avr_vcd_signal_t**

alias

irq

name

size

class pysimavr.swig.simavr.**avr_vcd_t**

avr

filename

log

logindex

output

period

signal

signal_count

start

class pysimavr.swig.simavr.**elf_firmware_t**

aref

avcc

bsssize

codeline

codesize

command_register_addr

console_register_addr

datasize

eeprom

eesize

flash

flashbase

flashsize

frequency

mmcu

trace

tracecount

tracename

traceperiod

vcc

class pysimavr.swig.simavr.**elf_firmware_t_trace**

addr

mask

name

5.2 high level interface

class pysimavr.ac.**Ac**(*avr*)

getirq(*pin*)

class pysimavr.avr.**Avr**(*firmware=None*, *mcu=None*, *f_cpu=None*, *avcc=5*, *vcc=5*)

avcc

fpeek(*addr*)

getirq(*pin*)

goto_cycle(*n*)

goto_time(*tsec*)

load_firmware(*firmware*)

move_time_marker(*tsec_diff*)

pause()

peek(*addr*)

reset()

run()

step(*n=1*, *sync=True*)

terminate()

time_passed()

vcc

exception pysimavr.avr.**UnkwownAvrError**

pysimavr.connect.**connect_irqs**(*irq_out*, *irq_in*, *bidirectional=False*) pysimavr.connect.**connect_pins_by_rule**(*rule*, *device_map*, *vcd=None*)

rule example:

B0 –> D4 -> vcd

B1 <== D5 B2 => D6 #B3 <=> D7

class pysimavr.firmware.**Firmware**(*filename=None*)

mcu

read(*filename*)

class pysimavr.inverter.**Inverter**(*avr*)

getirq(*pin*)

out(*i*)

class pysimavr.lcd.**Lcd**(*avr*, *size=(20*, *2)*)

get_char(*x*, *y*)

getirq(*pin*)

pinstate(*pin*)

reset()

class pysimavr.ledrow.**LedRow**(*avr*, *size=8*)

getirq(*pin*)

pinstate(*i*)

reset_dirty(*i*)

read and reset

class pysimavr.sgm7.**Sgm7**(*avr*, *size=4*)

digit_segments(*digit_index*)

getirq(*pin*)

pinindex(*pin_name*)

pinstate(*pin*)

reset_dirty(*digit_index*)

read and reset

class pysimavr.vcdfile.**VcdFile**(*avr*, *filename='gtkwave_output.vcd'*, *period=10*)

add_signal(*irq*, *name=None*, *bits=1*)

start()

stop()

terminate()

CHAPTER

SIX

DEVELOPMENT

6.1 Tools

1. setuptools
2. Paver
3. nose
4. ghp-import
5. pyflakes
6. pychecker
7. paved fork
8. Sphinx
9. sphinxcontrib-programscreenshot
10. sphinxcontrib-paverutils
11. `autorun` from sphinx-contrib (there is no simple method, you have to download/unpack/setup)

6.2 Install on ubuntu

```
sudo apt-get install python-setuptools
sudo apt-get install python-paver
sudo apt-get install python-nose
sudo easy_install ghp-import
sudo apt-get install pyflakes
sudo apt-get install pychecker
sudo easy_install https://github.com/ponty/paved/zipball/master
sudo apt-get install scrot
sudo apt-get install xvfb
sudo apt-get install xserver-xephyr
sudo apt-get install python-imaging
sudo apt-get install python-sphinx
sudo easy_install sphinxcontrib-programscreenshot
sudo easy_install sphinxcontrib-programoutput
sudo easy_install sphinxcontrib-paverutils
```

6.3 Tasks

Paver is used for task management, settings are saved in pavement.py. Sphinx is used to generate documentation.

print paver settings:

```
paver printoptions
```

clean generated files:

```
paver clean
```

generate documentation under *docs/_build/html*:

```
paver cog pdf html
```

upload documentation to github:

```
paver ghpages
```

run unit tests:

```
paver nose
#or
nosetests --verbose
```

check python code:

```
paver pyflakes
paver pychecker
```

generate python distribution:

```
paver sdist
```

upload python distribution to PyPI:

```
paver upload
```

CHAPTER

SEVEN

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX

p

pysimavr.ac, 13
pysimavr.avr, 13
pysimavr.connect, 13
pysimavr.firmware, 14
pysimavr.inverter, 14
pysimavr.lcd, 14
pysimavr.ledrow, 14
pysimavr.sgm7, 14
pysimavr.swig.ac_input, 7
pysimavr.swig.hd44780, 7
pysimavr.swig.inverter, 7
pysimavr.swig.ledrow, 8
pysimavr.swig.sgm7, 8
pysimavr.swig.simavr, 8
pysimavr.vcdfile, 14

INDEX

A

Ac (class in pysimavr.ac), 13
ac_input_t (class in pysimavr.swig.ac_input), 7
add_signal() (pysimavr.vcdfile.VcdFile method), 14
addr (pysimavr.swig.simavr.avr_symbol_t attribute), 9
addr (pysimavr.swig.simavr.elf_firmware_t_trace attribute), 13
alias (pysimavr.swig.simavr.avr_vcd_signal_t attribute), 12
aref (pysimavr.swig.simavr.avr_t attribute), 9
aref (pysimavr.swig.simavr.elf_firmware_t attribute), 12
avcc (pysimavr.avr.Avr attribute), 13
avcc (pysimavr.swig.simavr.avr_t attribute), 9
avcc (pysimavr.swig.simavr.elf_firmware_t attribute), 12
Avr (class in pysimavr.avr), 13
avr (pysimavr.swig.ac_input.ac_input_t attribute), 7
avr (pysimavr.swig.hd44780.hd44780_t attribute), 7
avr (pysimavr.swig.inverter.inverter_t attribute), 7
avr (pysimavr.swig.ledrow.ledrow_t attribute), 8
avr (pysimavr.swig.sgm7.sgm7_t attribute), 8
avr (pysimavr.swig.simavr.avr_io_t attribute), 8
avr (pysimavr.swig.simavr.avr_vcd_t attribute), 12
avr_io_t (class in pysimavr.swig.simavr), 8
avr_ioport_getirq_t (class in pysimavr.swig.simavr), 8
avr_ioport_state_t (class in pysimavr.swig.simavr), 8
avr_ioport_t (class in pysimavr.swig.simavr), 9
avr_irq_pool_t (class in pysimavr.swig.simavr), 9
avr_irq_t (class in pysimavr.swig.simavr), 9
avr_kind_t (class in pysimavr.swig.simavr), 9
avr_symbol_t (class in pysimavr.swig.simavr), 9
avr_t (class in pysimavr.swig.simavr), 9
avr_t_cycle_timer (class in pysimavr.swig.simavr), 11
avr_t_io (class in pysimavr.swig.simavr), 11
avr_t_io_r (class in pysimavr.swig.simavr), 11
avr_t_io_shared_io (class in pysimavr.swig.simavr), 11
avr_t_io_shared_io_io (class in pysimavr.swig.simavr), 11
avr_t_io_w (class in pysimavr.swig.simavr), 11
avr_trace_data_t (class in pysimavr.swig.simavr), 11
avr_trace_data_t_old (class in pysimavr.swig.simavr), 11
avr_vcd_log_t (class in pysimavr.swig.simavr), 11
avr_vcd_signal_t (class in pysimavr.swig.simavr), 12
avr_vcd_t (class in pysimavr.swig.simavr), 12

B

bit (pysimavr.swig.simavr.avr_ioport_getirq_t attribute), 8
bsssize (pysimavr.swig.simavr.elf_firmware_t attribute), 12

C

c (pysimavr.swig.simavr.avr_t_io_r attribute), 11
c (pysimavr.swig.simavr.avr_t_io_shared_io_io attribute), 11
c (pysimavr.swig.simavr.avr_t_io_w attribute), 11
codeend (pysimavr.swig.simavr.avr_t attribute), 9
codeline (pysimavr.swig.simavr.avr_trace_data_t attribute), 11
codeline (pysimavr.swig.simavr.elf_firmware_t attribute), 12
codesize (pysimavr.swig.simavr.elf_firmware_t attribute), 12
command_register_addr (pysimavr.swig.simavr.elf_firmware_t attribute), 12
connect_irqs() (in module pysimavr.connect), 13
connect_pins_by_rule() (in module pysimavr.connect), 13
console_register_addr (pysimavr.swig.simavr.elf_firmware_t attribute), 12
count (pysimavr.swig.simavr.avr_irq_pool_t attribute), 9
cursor (pysimavr.swig.hd44780.hd44780_t attribute), 7
cycle (pysimavr.swig.simavr.avr_t attribute), 9
cycle_timer (pysimavr.swig.simavr.avr_t attribute), 9
cycle_timer_map (pysimavr.swig.simavr.avr_t attribute), 10

D

data (pysimavr.swig.simavr.avr_t attribute), 10
datapins (pysimavr.swig.hd44780.hd44780_t attribute), 7
datasize (pysimavr.swig.simavr.elf_firmware_t attribute), 12
ddr (pysimavr.swig.simavr.avr_ioport_state_t attribute), 8

dealloc (pysimavr.swig.simavr.avr_io_t attribute), 8
digit_count (pysimavr.swig.sgm7.sgm7_t attribute), 8
digit_pin (pysimavr.swig.sgm7.sgm7_t attribute), 8
digit_port (pysimavr.swig.sgm7.sgm7_t attribute), 8
digit_segments (pysimavr.swig.sgm7.sgm7_t attribute), 8
digit_segments() (pysimavr.sgm7.Sgm7 method), 14
digit_segments_changed (pysimavr.swig.sgm7.sgm7_t attribute), 8

E

e2end (pysimavr.swig.simavr.avr_t attribute), 10
eeprom (pysimavr.swig.simavr.elf_firmware_t attribute), 12
eesize (pysimavr.swig.simavr.elf_firmware_t attribute), 12
eind (pysimavr.swig.simavr.avr_t attribute), 10
elf_firmware_t (class in pysimavr.swig.simavr), 12
elf_firmware_t_trace (class in pysimavr.swig.simavr), 13

F

filename (pysimavr.swig.simavr.avr_vcd_t attribute), 12
Firmware (class in pysimavr.firmware), 14
flags (pysimavr.swig.hd44780.hd44780_t attribute), 7
flags (pysimavr.swig.simavr.avr_irq_t attribute), 9
flash (pysimavr.swig.simavr.avr_t attribute), 10
flash (pysimavr.swig.simavr.elf_firmware_t attribute), 12
flashbase (pysimavr.swig.simavr.elf_firmware_t attribute), 12
flashend (pysimavr.swig.simavr.avr_t attribute), 10
flashsize (pysimavr.swig.simavr.elf_firmware_t attribute), 12
fpeek() (pysimavr.avr.Avr method), 13
frequency (pysimavr.swig.simavr.avr_t attribute), 10
frequency (pysimavr.swig.simavr.elf_firmware_t attribute), 12
fuse (pysimavr.swig.simavr.avr_t attribute), 10

G

gdb (pysimavr.swig.simavr.avr_t attribute), 10
gdb_port (pysimavr.swig.simavr.avr_t attribute), 10
get_char() (pysimavr.lcd.Lcd method), 14
getirq() (pysimavr.ac.Ac method), 13
getirq() (pysimavr.avr.Avr method), 13
getirq() (pysimavr.inverter.Inverter method), 14
getirq() (pysimavr.lcd.Lcd method), 14
getirq() (pysimavr.ledrow.LedRow method), 14
getirq() (pysimavr.sgm7.Sgm7 method), 14
goto_cycle() (pysimavr.avr.Avr method), 13
goto_time() (pysimavr.avr.Avr method), 13

H

h (pysimavr.swig.hd44780.hd44780_t attribute), 7
hd44780_t (class in pysimavr.swig.hd44780), 7
hook (pysimavr.swig.simavr.avr_irq_t attribute), 9

I

i_shadow (pysimavr.swig.simavr.avr_t attribute), 10
init (pysimavr.swig.simavr.avr_t attribute), 10
Inverter (class in pysimavr.inverter), 14
inverter_t (class in pysimavr.swig.inverter), 7
io (pysimavr.swig.simavr.avr_ioport_t attribute), 9
io (pysimavr.swig.simavr.avr_t attribute), 10
io (pysimavr.swig.simavr.avr_t_io_shared_io attribute), 11
io_port (pysimavr.swig.simavr.avr_t attribute), 10
io_shared_io (pysimavr.swig.simavr.avr_t attribute), 10
io_shared_io_count (pysimavr.swig.simavr.avr_t attribute), 10
ioctl (pysimavr.swig.simavr.avr_io_t attribute), 8
irq (pysimavr.swig.ac_input.ac_input_t attribute), 7
irq (pysimavr.swig.hd44780.hd44780_t attribute), 7
irq (pysimavr.swig.inverter.inverter_t attribute), 7
irq (pysimavr.swig.ledrow.ledrow_t attribute), 8
irq (pysimavr.swig.sgm7.sgm7_t attribute), 8
irq (pysimavr.swig.simavr.avr_io_t attribute), 8
irq (pysimavr.swig.simavr.avr_ioport_getirq_t attribute), 8
irq (pysimavr.swig.simavr.avr_irq_pool_t attribute), 9
irq (pysimavr.swig.simavr.avr_irq_t attribute), 9
irq (pysimavr.swig.simavr.avr_t_io attribute), 11
irq (pysimavr.swig.simavr.avr_vcd_signal_t attribute), 12
irq_count (pysimavr.swig.simavr.avr_io_t attribute), 8
irq_ioctl_get (pysimavr.swig.simavr.avr_io_t attribute), 8
irq_names (pysimavr.swig.simavr.avr_io_t attribute), 8
irq_pool (pysimavr.swig.simavr.avr_t attribute), 10

K

kind (pysimavr.swig.simavr.avr_io_t attribute), 8

L

Lcd (class in pysimavr.lcd), 14
LedRow (class in pysimavr.ledrow), 14
ledrow_t (class in pysimavr.swig.ledrow), 8
load_firmware() (pysimavr.avr.Avr method), 13
log (pysimavr.swig.simavr.avr_vcd_t attribute), 12
logindex (pysimavr.swig.simavr.avr_vcd_t attribute), 12

M

make (pysimavr.swig.simavr.avr_kind_t attribute), 9
mask (pysimavr.swig.simavr.elf_firmware_t_trace attribute), 13
mcu (pysimavr.firmware.Firmware attribute), 14
mmcu (pysimavr.swig.simavr.avr_t attribute), 10
mmcu (pysimavr.swig.simavr.elf_firmware_t attribute), 13
move_time_marker() (pysimavr.avr.Avr method), 13

N

name (pysimavr.swig.simavr.avr_ioport_state_t attribute), 8
name (pysimavr.swig.simavr.avr_ioport_t attribute), 9
name (pysimavr.swig.simavr.avr_irq_t attribute), 9
name (pysimavr.swig.simavr.avr_vcd_signal_t attribute), 12
name (pysimavr.swig.simavr.elf_firmware_t_trace attribute), 13
names (pysimavr.swig.simavr.avr_kind_t attribute), 9
next (pysimavr.swig.simavr.avr_io_t attribute), 8
next_cycle_timer (pysimavr.swig.simavr.avr_t attribute), 10

O

old (pysimavr.swig.simavr.avr_trace_data_t attribute), 11
old_pci (pysimavr.swig.simavr.avr_trace_data_t attribute), 11
out (pysimavr.swig.inverter.inverter_t attribute), 7
out() (pysimavr.inverter.Inverter method), 14
output (pysimavr.swig.simavr.avr_vcd_t attribute), 12

P

param (pysimavr.swig.simavr.avr_t_cycle_timer attribute), 11
param (pysimavr.swig.simavr.avr_t_io_r attribute), 11
param (pysimavr.swig.simavr.avr_t_io_shared_io_io attribute), 11
param (pysimavr.swig.simavr.avr_t_io_w attribute), 11
pause() (pysimavr.avr.Avr method), 13
pc (pysimavr.swig.simavr.avr_t attribute), 10
pc (pysimavr.swig.simavr.avr_trace_data_t_old attribute), 11
pcint (pysimavr.swig.simavr.avr_ioport_t attribute), 9
peek() (pysimavr.avr.Avr method), 13
pending (pysimavr.swig.simavr.avr_t attribute), 10
pending_wait (pysimavr.swig.simavr.avr_t attribute), 10
period (pysimavr.swig.simavr.avr_vcd_t attribute), 12
pin (pysimavr.swig.simavr.avr_ioport_state_t attribute), 9
pinindex() (pysimavr.sgm7.Sgm7 method), 14
pinstate (pysimavr.swig.hd44780.hd44780_t attribute), 7
pinstate (pysimavr.swig.ledrow.ledrow_t attribute), 8
pinstate (pysimavr.swig.sgm7.sgm7_t attribute), 8
pinstate() (pysimavr.lcd.Lcd method), 14
pinstate() (pysimavr.ledrow.LedRow method), 14
pinstate() (pysimavr.sgm7.Sgm7 method), 14
pinstate_changed (pysimavr.swig.ledrow.ledrow_t attribute), 8
pool (pysimavr.swig.simavr.avr_irq_t attribute), 9
port (pysimavr.swig.simavr.avr_ioport_state_t attribute), 9
pysimavr.ac (module), 13
pysimavr.avr (module), 13
pysimavr.connect (module), 13
pysimavr.firmware (module), 14
pysimavr.inverter (module), 14
pysimavr.lcd (module), 14
pysimavr.ledrow (module), 14
pysimavr.sgm7 (module), 14
pysimavr.swig.ac_input (module), 7
pysimavr.swig.hd44780 (module), 7
pysimavr.swig.inverter (module), 7
pysimavr.swig.ledrow (module), 8
pysimavr.swig.sgm7 (module), 8
pysimavr.swig.simavr (module), 8
pysimavr.vcdfile (module), 14

R

r (pysimavr.swig.simavr.avr_t_io attribute), 11
r_ddr (pysimavr.swig.simavr.avr_ioport_t attribute), 9
r_pcint (pysimavr.swig.simavr.avr_ioport_t attribute), 9
r_pin (pysimavr.swig.simavr.avr_ioport_t attribute), 9
r_port (pysimavr.swig.simavr.avr_ioport_t attribute), 9
ramend (pysimavr.swig.simavr.avr_t attribute), 10
rampz (pysimavr.swig.simavr.avr_t attribute), 10
read() (pysimavr.firmware.Firmware method), 14
readpins (pysimavr.swig.hd44780.hd44780_t attribute), 7
reset (pysimavr.swig.simavr.avr_io_t attribute), 8
reset (pysimavr.swig.simavr.avr_t attribute), 10
reset() (pysimavr.avr.Avr method), 13
reset() (pysimavr.lcd.Lcd method), 14
reset_dirty() (pysimavr.ledrow.LedRow method), 14
reset_dirty() (pysimavr.sgm7.Sgm7 method), 14
run (pysimavr.swig.simavr.avr_t attribute), 10
run() (pysimavr.avr.Avr method), 13

S

segment_pin (pysimavr.swig.sgm7.sgm7_t attribute), 8
segment_port (pysimavr.swig.sgm7.sgm7_t attribute), 8
Sgm7 (class in pysimavr.sgm7), 14
sgm7_t (class in pysimavr.swig.sgm7), 8
signal (pysimavr.swig.simavr.avr_vcd_log_t attribute), 12
signal (pysimavr.swig.simavr.avr_vcd_t attribute), 12
signal_count (pysimavr.swig.simavr.avr_vcd_t attribute), 12
signature (pysimavr.swig.simavr.avr_t attribute), 10
size (pysimavr.swig.simavr.avr_vcd_signal_t attribute), 12
sleep (pysimavr.swig.simavr.avr_t attribute), 10
sp (pysimavr.swig.simavr.avr_trace_data_t_old attribute), 11
special_deinit (pysimavr.swig.simavr.avr_t attribute), 10
special_init (pysimavr.swig.simavr.avr_t attribute), 10
sreg (pysimavr.swig.simavr.avr_t attribute), 10
start (pysimavr.swig.simavr.avr_vcd_t attribute), 12
start() (pysimavr.vcdfile.VcdFile method), 14
state (pysimavr.swig.simavr.avr_t attribute), 10
step() (pysimavr.avr.Avr method), 13

stop() (pysimavr.vcdfile.VcdFile method), 14
symbol (pysimavr.swig.simavr.avr_symbol_t attribute), 9

T

terminate() (pysimavr.avr.Avr method), 13
terminate() (pysimavr.vcdfile.VcdFile method), 14
time_passed() (pysimavr.avr.Avr method), 13
timer (pysimavr.swig.simavr.avr_t_cycle_timer attribute), 11
touched (pysimavr.swig.simavr.avr_trace_data_t attribute), 11
trace (pysimavr.swig.simavr.avr_t attribute), 10
trace (pysimavr.swig.simavr.elf_firmware_t attribute), 13
trace_data (pysimavr.swig.simavr.avr_t attribute), 10
tracecount (pysimavr.swig.simavr.elf_firmware_t attribute), 13
tracename (pysimavr.swig.simavr.elf_firmware_t attribute), 13
traceperiod (pysimavr.swig.simavr.elf_firmware_t attribute), 13

U

UnkwownAvrError, 13
used (pysimavr.swig.simavr.avr_t_io_shared_io attribute), 11

V

value (pysimavr.swig.ac_input.ac_input_t attribute), 7
value (pysimavr.swig.simavr.avr_irq_t attribute), 9
value (pysimavr.swig.simavr.avr_vcd_log_t attribute), 12
vcc (pysimavr.avr.Avr attribute), 13
vcc (pysimavr.swig.simavr.avr_t attribute), 10
vcc (pysimavr.swig.simavr.elf_firmware_t attribute), 13
vcd (pysimavr.swig.simavr.avr_t attribute), 10
VcdFile (class in pysimavr.vcdfile), 14
vector (pysimavr.swig.simavr.avr_t attribute), 11
vector_size (pysimavr.swig.simavr.avr_t attribute), 11
vram (pysimavr.swig.hd44780.hd44780_t attribute), 7

W

w (pysimavr.swig.hd44780.hd44780_t attribute), 7
w (pysimavr.swig.simavr.avr_t_io attribute), 11
when (pysimavr.swig.simavr.avr_t_cycle_timer attribute), 11
when (pysimavr.swig.simavr.avr_vcd_log_t attribute), 12